

Installation on OS/390 or MVS

Software AG recommends that you keep unmodified copies of all materials distributed or created as part of the installation process. This may assist with problem diagnosis later by providing an untouched sample of any given item.

Note:

Additional installation steps for Natural CGI are documented in the chapter Installing Natural CGI

This document covers the following topics:

- The Installation Tape
 - Creating the Mainframe Datasets
 - Running the Installation Jobs
 - Installing under the SMARTS Server Environment
 - Installing under Com-plete
 - Where Next ?
-

The Installation Tape

Tape Contents

Datasets

The following table lists the product datasets, what the dataset contains, and how it is created. While you are free to rename the datasets, the dataset names used in the table are used consistently throughout the product documentation to ensure clarity.

Distributed Datasets

Dataset	Contains ...
HTPvrs.LOAD	all load modules required by SMARTS HTTP server
HTPvrs.SRCE	all sample source members and macros
HTPvrs.JOBS	all sample JCL required
HTPvrs.INPL	an INPL file for Natural modules and example programs
HTPvrs.UPDW	an INPL update for Natural Web Interface compatibility
HTPvrs.GIFS	GIF files for the Natural Web Interface example demo

Datasets Created during the Installation Process

Name	Dataset containing ...
HTPvrs.USER.SRCE	source members
HTPvrs.USER.LOAD	load modules for all SMARTS environments

Creating the PC Files

Step 1: Copy the installation files to disk

- Copy all of the installation files to a directory on a hard disk where a minimum of 4MB of disk space must be available on a temporary basis in addition to the installation files.

The files are called HTPvrs#n.EXE where 'vrs' is the version, revision, and system maintenance level of SMARTS and 'n' is a sequential number depending on the number of files provided.

All files must be copied from the installation media.

Step 2: Execute each file

- Execute each file provided which will expand to create one or more new files in the same directory.

The files created are listed in the following table along with a description of their contents:

File Name	Contents
\$ALLOC	Sample JCL to allocate the necessary datasets on the mainframe into which the PC datasets will be uploaded. This job is referred to later in this document.
\$TSORECV	Sample JCL to create the actual SMARTS installation datasets from the uploaded datasets. This job is referred to later in this document.
LOAD	HTPvrs.LOAD in off-loaded format.
SOURCE	HTPvrs.SRCE in off-loaded format.
JOBS	HTPvrs.JOBS in off-loaded format.
INPL	HTPvrs.INPL in off-loaded format
UPDW	HTPvrs.UPDW in off-loaded format
GIFS	HTPvrs.GIFS in off-loaded format.

Creating the Mainframe Datasets

Step 1: Allocate the Datasets

- Allocate a dataset with the following DCB information for each off-loaded file that now exists on the PC:

RECFM=FB
 LRECL=80
 BLKSIZE=3120

Software AG recommends that you

- name each dataset you allocate based on its HTPvrs.* name with the suffix .SEQ. For example, the dataset for the PC LOAD file would be HTPvrs.LOAD.SEQ.
- allocate the datasets in blocks. You can determine the number of blocks required by dividing the size of the PC file in bytes by the blocksize 3120 and adding 1. For example, if the PC file is 3,480 bytes, allocate 2 blocks.

You may allocate the datasets using either

- TSO; or
- the \$ALLOC file, which may be uploaded to a source dataset on the mainframe as a text file (using ASCII/EBCDIC translation) and modified to suit the installation requirements and to reflect the correct space allocation required for each dataset.

Step 2: Upload the Data

- Once you have allocated the sequential datasets on the mainframe, use a binary transfer to upload the binary files to their equivalent mainframe dataset.

IND\$FILE and standard FTP implementations all offer the binary transfer capability.

The following table pairs the PC file with the mainframe dataset to which it should be loaded:

PC File	Mainframe Dataset
LOAD	HTPvrs.LOAD.SEQ
SRCE	HTPvrs.SRCE.SEQ
JOBS	HTPvrs.JOBS.SEQ
INPL	HTPvrs.INPL.SEQ
UPDW	HTPvrs.UPDW.SEQ
GIFS	HTPvrs.GIFS.SEQ

Step 3: Create the SMARTS Datasets

- Once you have uploaded the data to the mainframe, the actual mainframe installation datasets must be created. Because the sequential datasets uploaded from the PC are actually the result of a TSO TRANSMIT command, the datasets must be recreated using a TSO RECEIVE command.

You may either

- upload the \$TSORECV file on the PC to a source dataset on the mainframe as a text file (using ASCII/EBCDIC translation) and use it as a sample to issue the appropriate commands in batch; or

- RECEIVE the datasets individually using TSO commands.

Software AG recommends that RECEIVE be allowed to allocate the space required by the datasets as it can determine what is required from internal information in the sequential file itself.

Once the SMARTS datasets have been created, delete the sequential datasets created for the data uploaded from the PC (that is, the datasets with the .SEQ suffix). These can be created again from the PC files, if necessary.

Running the Installation Jobs

The following procedure installs the SMARTS HTTP server product.

Step 1: Allocate and Initialize User PDS Datasets

- To create and initialize the datasets required for the SMARTS HTTP server, modify the sample job in member \$JCLALLO on the HTPvrs.SRCE dataset to suit your installation's environment, and run it to create the appropriate datasets.

This job also copies all modifiable members from the HTPvrs.SRCE dataset to the newly created HTPvrs.USER.SRCE dataset in order to retain all HTPvrs.SRCE members as delivered.

Note:

To ensure that the \$JCLALLO member remains as delivered on the HTPvrs.SRCE dataset for future reference, Software AG recommends that you modify and submit the job from the editor without saving it. Once the job has completed successfully, the job may be saved in the HTPvrs.USER.SRCE dataset.

Step 2: Install the HTTP Server HLL Interface Modules

- A number of high-level language interface modules must be linked in order to execute the HTTP server.

Modify and run the sample JCL member HJENLINK in the HTPvrs.JOBS dataset to link the modules in the appropriate way to the HTPvrs.USER.LOAD dataset.

Ensure that the job finishes with condition code "0".

Step 3: Install the Natural INPL File

-

Note:

This step applies only if you are running with Natural.

Install the INPL file delivered with the SMARTS HTTP server creating a Natural library called HTPvrs.

Refer to the chapter Installing Natural CGI for information about installing Natural CGI support.

Step 4: Install the Natural INPL Update File

-

Note:

This step applies only if you are running with Natural 3.1 or above and the Natural Web Interface.

Install the INPL update file HTPvrs.UPDW delivered with SMARTS to update the SYSWEB library.

Refer to the chapter Installing Natural CGI for information about installing Natural Web Interface support.

Step 5: Customize the HTTP Server

- The sample configuration member HAANCONF was copied into the HTPvrs.USER.SRCE dataset during installation. Software AG recommends that you use the member as delivered for the installation verification routines unless the default port number (8080) is inappropriate.

If the HTTP server is to be used with Natural, it may be necessary to change the NATTHRD and NATLIB parameters.

If it is necessary to change the port number or any other parameter (the default port is 8080 and is already set in the supplied default HTTP configuration):

1. Modify the parameters in HAANCONF.
2. Recompile HAANCONF using the sample HJBNA CNF JCL in HTPvrs.JOBS.
3. Assemble HAANCONF to the HTPvrs.USER.LOAD library available to the server at startup.

Refer to the HTTP Server Use and Customization chapter later in this manual for details about this process and the parameters that can be specified.

Step 6: Customize the SMARTS Environment

- The SMARTS environment configuration member PXANCONF must include the communication driver interface (CDI) protocol definition for cgistdio.

Following is a sample of the CDI_DRIVER parameter specification:

```
CDI_DRIVER=( 'cgistdio,HAANPCGI' )
```

See the SMARTS Installation and Operations Manual for more information about this step.

Installing under the SMARTS Server Environment

Step 1: Modify the SMARTS Server Start-up Procedure or Job

- Include the following datasets in the COMPLIB dataset concatenation in the order shown:

```
// DD DISP=SHR,DSN=HTPvrs.USER.LOAD
// DD DISP=SHR,DSN=HTPvrs.LOAD
```

You may optionally add the following DD statement in the JCL to direct output to a specific job class or dataset:

HTPTRCE	the HTTP server trace output when HTTP server tracing is active
---------	---

Step 2: Modify the SMARTS Server Start-up Parameters

- The member HJENPARM on the HTPvrs.JOBS dataset provides a sample set of the parameters required by the SMARTS server to
 - start the HTTP server; and
 - define the various interface extensions as RESIDENTPAGE.

Add the parameters in HJENPARM into your standard set.

At least one thread with 404 kilobytes defined below the 16 megabyte line and 1 megabyte above the line is required to run the installation verification programs. The values are not absolute and may be reduced depending on the server usage and the language environment configuration.

Installing under Com-plete

Note:

The steps in this section apply only if you are running the SMARTS HTTP server under Com-plete.

The procedure described in this section installs the SMARTS HTTP server under Com-plete.

Step 1: Modify the Com-plete Start-up Procedure or Job

- Include the datasets in the COMPLIB dataset concatenation as for the SMARTS server environment.

Step 2: Modify the Com-plete Start-up Parameters

- Add the parameters from the sample member HJENPARM to the standard set as for the SMARTS server environment.

Step 3: Catalog PAENSTRT

- Use the ULIB utility of Com-plete to catalog the PAENSTRT program with an initial thread size of 400 kilobytes.

Step 4: Install the LE in Com-plete

- To test the C, PL/I, and COBOL programs delivered with SMARTS, you must be able to run language environment (LE)-enabled programs in the Com-plete system.

Refer to the Com-plete documentation for information.

Step 5: Verify the Installation

- See the chapter Verifying the Installation.

Execute the steps for the SMARTS server environment against Com-plete to ensure that SMARTS is running successfully under Com-plete.

In addition to running the programs using the HTTP server, it should be possible to execute the programs from the command line of a Com-plete session. The output appears in the stdout file.

Step 6: Restart Com-plete

- Near the end of initialization processing, messages are issued to the console indicating that the HTTP server has been started.

If the server does not start successfully, check for error messages and verify the installation steps again.

Where Next ?

Continue to the chapter Verifying the Installation for information about verifying the SMARTS HTTP server installation and troubleshooting.

Then familiarize yourself with the customization and configuration options available in the product. Following the customization sections are a number of sections detailing specific functionality and how to implement this functionality in the SMARTS environment.

For specific information about the programming interfaces and how to use them, refer to the chapter Programming CGI Requests later in this manual.